

Input reading: 3pts | Initialization: 3 pts | Loop condition: 4 pts | Counters logic: 4 pts | Stop conditions: 3 | Final output: 3

include <stdio.h>

int main () {

int A, N, S, a;

int x=0, i=0, p=0;

printf("Enter the integer positive N: ");

scanf("%d", &N);

printf("Enter the minimum attendance required A: ");

scanf("%d", &A);

printf("Enter absence threshold S: ");

scanf("%d", &S);

printf("Enter the number of attended sessions x: ");

scanf("%d", &x);

if (x < A) {

printf("The student is Absent: ");

else

printf("The student is present: ");

}

}

while (x >= A) {

p = p + 1;

printf("%d", p);

a = N - p;

printf("%d", a);

// توقف الطالب في إحدى الحالتين

if (N == p) {

printf("إيقاف الطالب");

if (a == S) {

printf("إيقاف الطالب");

if (p >= A) {

printf("إيقاف الطالب");

else

printf("إيقاف الطالب");

}

}

return 0;

}

Copy 1

```
#include <stdio.h>
int main()
{
    int A, N, S, a;
    int x, i = 0, p = 0;
    printf("Enter the integer positive N: ");
    scanf("%d", &N);
    printf("Enter the minimum attendance required A: ");
    scanf("%d", &A);
    printf("Enter absence threshold S: ");
    scanf("%d", &S);
    printf("Enter the number of attended sessions x: ");
    scanf("%d", &x);
    if (x < A)
    {
        printf("the student is absent");
    }

    else
    {
        printf("the student is present");
    }

    while (x >= A)
    {
        p = p + 1;
        printf("%d", p);
        a = N - p;
    }

    printf("%d", A);
    if (p > A)
    {
        printf("Exam valid");
    }

    else
    {
        printf("Exam cancelled");
    }

    return 0;
}
```

Analyse :**Algorithmique :**

- Lecture correcte de N, A, S.
- Lecture de x **AVANT** la boucle while. La variable x ne change jamais dans la boucle ! Boucle infinie si $x \geq A$ ou sautée si $x < A$.
- Corps de boucle : $a = N - p$. Logique confuse.
- Condition d'arrêt while $(x \geq A)$ n'a aucun sens par rapport à l'énoncé (fin si tous traités ou seuil absents).

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	OK.
Initialisation	3 / 3	OK.
Condition boucle	0 / 4	Incorrecte et boucle potentiellement infinie.
Logique prés./abs.	2 / 4	Test correct, mais mal placé (hors itération).
Compteurs	1 / 3	Incrémentation présente mais logique fausse.
Affichages inter.	1 / 2	Présents mais données fausses.
Affichage final	1 / 1	OK.

NOTE FINALE : 11 / 20**Feedback :**

- **Appréciation globale : Moyen -.** Erreur classique de débutant : lecture de la variable de contrôle hors de la boucle.
-

Input reading: 3pts | Initialization: 3 pts | Loop condition: 4 pts | Counters logic: 4 pts | Stop conditions: 3 | Final output: 3

```

#include <stdio.h>
int main()
{
    int N, A, S, X;
    printf("Enter Number of attended Sessions :");
    int i = 0, j = 0;
    while (i <= N || j <= S)
    {
        j++;
        i++;
        if (scanf("%d", &X) <= A)
        {
            printf("Number of Student Absent is %d", j);
        }
        else { printf("Number of Student Present is %d", i); }
    }
    b = N - i;
    c = N - j;
    printf("Number of All Student Absent is : %d", c);
    printf("Number of All Student Present is : %d", b);
    if (c >= S)
    {
        printf("Session cancelled");
    }
    else { printf("Session Vailed"); }
    Return 0;
}

```

Copy 2

```
#include <stdio.h>
int main()
{
    int N, A, S, X;
    printf("Enter Number of attended sessions: ");
    int i = 0, j = 0;
    while (i <= N || j <= S)
    {
        j++;
        i++;
        if (scanf("%d", &X) <= A)
        {
            printf("Number of Student Absent is %d", j);
        }

        else
        {
            printf("Number of Student Present is %d", i);
        }

    }

    b = N - i;
    c = N - j;
    printf("Number of All Student Absent is: %d", c);
    printf("Number of All Student Present is: %d", b);
    if (c >= S)
    {
        printf("Session cancelled");
    }

    else
    {
        printf("Session Valid");
    }

    return 0;
}
```

Analyse :**Algorithmique :**

- Lecture X dans la condition du while ? `if ( scanf ( . . . ) <= A )`. C'est astucieux mais techniquement `scanf` renvoie le nombre d'éléments lus, pas la valeur ! Donc `scanf` renvoie 1, qui est comparé à A. Si $A > 1$, résultat toujours vrai.
- Boucle while `( i <= N || j <= S )` : OU logique au lieu de ET. Continue tant qu'une limite n'est pas atteinte (risque dépassement).
- Calculs finaux `b = N - i` corrects sur le principe.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	0 / 3	Absente (un <code>printf</code> demande mais pas de <code>scanf</code> pour N, A, S !!). Variables init à 0 dans déclarations pseudo <code>printf(N,A,S)</code> .
Initialisation	2 / 3	Variables init.
Condition boucle	2 / 4	Erreur logique <code> </code> .
Logique prés./abs.	1 / 4	Erreur majeure sur utilisation de <code>scanf</code> en expression.
Compteurs	2 / 3	OK.
Affichages inter.	2 / 2	OK.
Affichage final	1 / 1	OK.

NOTE FINALE : 08 / 20**Feedback :**

- **Appréciation globale : Insuffisant.** Pas de lecture des paramètres initiaux. Confusion sur le retour de `scanf`.
-


```
#include <stdio.h>
```

```
int main () {
```

```
int N, A, S, Np, Na
```

```
int i = 0;
```

```
while (i = < N , i < S , i++) do
```

```
    i ← i + 1
```

```
    Printf("the student number is: ", Student number);
```

```
    scanf("%d", &x);
```

```
    if (x < A) {
```

```
        Printf("the student is absent");
```

```
        Na ← N - A
```

```
        Printf("number of absent student is:", Na);
```

```
    else
```

```
        Printf("the student is present");
```

```
        Np = N - S
```

```
        Printf("number of Present student is:", Np);
```

```
    end if
```

```
end while
```

```
if (present students > absent students) {
```

```
    Printf("session valid");
```

```
    Else
```

```
        Printf("session cancelled");
```

```
end if
```

```
return 0;
```

Copy 3

```
#include <stdio.h>
int main()
{
    int N, A, S, Np, Na;
    int i = 0;
    while (i <= N, i <S, i++)
    {
        i = i + 1;
        printf("the student number is: ", student_number);
        scanf("%d", &x);
        if (x <A)
        {
            printf("the student is absent");
            Na = N - A;
            printf("number of absent student is: ", Na);
        }

        else
        {
            printf("the student is present");
            Np = N - S;
            printf("number of present student is: ", Np);
        }

    }

    if (present_students> absent_students)
    {
        printf("session valid");
    }

    else
    {
        printf("session cancelled");
    }

    return 0;
}
```


Analyse :**Algorithmique :**

- Boucle while (`i <= N, i < S, i++`) : Syntaxe for utilisée dans un while ? Opérateur virgule en C évalue tout et renvoie le dernier. Condition `i++` est toujours vraie (sauf overflow). Boucle infinie probable ou syntax error.
- Variables non initialisées `N, A, S`. Pas de lecture.
- Logique interne correcte (`if (x < A)`).

Notation :

Critère	Points	Commentaire
Lecture <code>N, A, S</code>	0 / 3	Absente.
Initialisation	0 / 3	Variables non init.
Condition boucle	0 / 4	Syntaxe invalide.
Logique prés./abs.	3 / 4	Correcte.
Compteurs	0 / 3	Calculs mathématiques au lieu d'incrémentation.
Affichages inter.	1 / 2	OK.
Affichage final	1 / 1	OK.

NOTE FINALE : 05 / 20**Feedback :**

- **Appréciation globale : Très Insuffisant.** Code non fonctionnel.
-

```

#include <stdio.h>
#include <stdlib.h>
int main() {
    int N;
    int A, S;
    int id;
    printf("entre number of registered students:");
    scanf("%d", &N);
    printf("minimum attendance required:");
    scanf("%d", &A);
    printf("absence threshold:");
    scanf("%d", &S);
    for(int i=1; i<=N; i++) {
        id = i + 1;
        printf("%d", id);
        scanf("%d", &x);
        if (x < A) {
            printf("student is absent");
        } else {
            printf("student is present");
        }
        if (student absent = S) { // توقف الكلاس
            printf("session cancelled");
        } else {
            printf("session valid");
        }
    }
    return 0;
}

```

Copy 4

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int N, A, S;
    int id;
    printf("entre number of registered students: ");
    scanf("%d", &N);
    printf("minimum attendance required: ");
    scanf("%d", &A);
    printf("absence threshold: ");
    scanf("%d", &S);
    for (int i = 1; i <= N; i++)
    {
        id = i + 1;
        printf("%d", id);
        scanf("%d", &x);
        if (x < A)
        {
            printf("student is absent");
        }

        else
        {
            printf("student is present");
        }

    }

    if (student_absent == S)
    {
        printf("session cancelled");
    }

    else
    {
        printf("session valid");
    }

    return 0;
}
```

Analyse :

Algorithmique :

- Lectures correctes.
- Boucle for sur N. Manque condition d'arrêt sur le seuil S.
- Logique interne correcte.
- Mais **aucun compteur** n'est incrémenté ! `student_absent` utilisé à la fin n'est jamais calculé.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	OK.
Initialisation	1 / 3	Manque compteurs.
Condition boucle	2 / 4	Partielle (S ignoré).
Logique prés./abs.	3 / 4	Bon test.
Compteurs	0 / 3	Absents.
Affichages inter.	1 / 2	OK.
Affichage final	0 / 1	Variable inexistante.

NOTE FINALE : 10 / 20

Feedback :

- **Appréciation globale : Moyen -.** Structure ok mais oubli fonctionnel majeur (les compteurs).
-

Input reading: 3pts | Initialization: 3 pts | Loop condition: 4 pts | Counters logic: 4 pts | Stop conditions: 3 | Final output: 3

#include <stdio.h>

int main () {

int A;

printf("entre A\n");

scanf("%d", &A);

int N;

printf("entre N");

scanf("%d", &N);

int S;

printf("entre S");

scanf("%d", &S);

int x, Absent, Present;

int i, j;

for (i = 0; i <= N; i++) { scanf("%d", &x);

printf("x %d = \n", i);

if (x < A) {

printf("x is Absent \n", Absent);

Absent++;

else {

printf("x is Present \n", Present);

Present++;

printf("the Absent is: %d \n", Absent);

printf("the Present is: %d \n", Present);

if (Absent < S) {

printf(" ok \n");

}

else {

printf(" no \n");

}

}

return 0;

}

Copy 5

```
#include <stdio.h>
int main()
{
    int A;
    printf("entre A\n");
    scanf("%d", &A);
    int N;
    printf("entre N\n");
    scanf("%d", &N);
    int S;
    printf("entre S\n");
    scanf("%d", &S);
    int x, Absent, Present;
    int i, j;
    for (i = 0; i <= N; i++)
    {
        scanf("%d", &x);
        printf("x: %d", i);
        if (x < A)
        {
            printf("x is absent", Absent);
            Absent++;
        }

        else
        {
            printf("x is present", Present);
            Present++;
        }

    }

    printf("the Absent is: %d", Absent);
    printf("the Present is: %d", Present);
    if (Absent < S)
    {
        printf("valid");
    }

    else
    {
        printf("cancelled");
    }

    return 0;
}
```

Analyse :

- Algorithmique :**
- Lectures correctes.
 - Boucle for sur N.
 - Logique correcte.
 - Compteurs incrémentés mais **non initialisés** (Absent, Present).
 - Condition d'arrêt sur S manquante dans la boucle (testée seulement à la fin).

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	OK.
Initialisation	0 / 3	Non initialisées.
Condition boucle	2 / 4	Partielle.
Logique prés./abs.	4 / 4	OK.
Compteurs	3 / 3	OK (sauf init).
Affichages inter.	1 / 2	OK.
Affichage final	1 / 1	OK.

NOTE FINALE : 14 / 20

Feedback :

- **Appréciation globale : Moyen.** Attention à l'initialisation des variables locales en C (valeur indéterminée).
-


```
#include <stdio.h>

int main() {
    int N, A, S, x, i;
    printf("entre N, A, S");
    scanf("%d", &N);
    scanf("%d", &A);
    scanf("%d", &S);
```

```
for (i = 1; i <= N, i++) {
    printf("entre x");
    scanf("%d", &x);
    if (x < A) {
        printf("طالب غائب");
    }
    else {
        printf("طالب حاضري");
    }
}
```

$i = i + 1$;

عدد الطلبة مسنور = $x + i$;

عدد الطلبة غائب = $S + i$;

if (عدد الطلبة مسنور > عدد الطلبة غائب)

```
{
    printf("الامتحان صالح");
}
```

```
else
{
```

```
printf("الامتحان ملغى");
}
```

```
return 0;
```

```
}
```

Copy 6

```
#include <stdio.h>
int main()
{
    int N, A, S, x, i;
    printf("entre N, A, S: ");
    scanf("%d %d %d", &N, &A, &S);
    for (i = 1; i <= N; i++)
    {
        printf("entre x : ");
        scanf("%d", &x);
        if (x < A)
        {
            printf("student absent");
        }

        else
        {
            printf("student present");
        }

        i = i + 1;
        if (absent_count >= S)
        {
            printf("Exam valid");
        }

        else
        {
            printf("Exam cancelled");
        }

        return 0;
    }
}
```

Analyse :

- Algorithmique :**
- Lectures correctes.
 - Boucle for : return 0 à la fin de la première itération. Le programme s'arrête tout de suite.
 - Utilise absent_count non déclaré.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	OK.
Condition boucle	1 / 4	Interrompue.
Logique prés./abs.	2 / 4	OK.
Compteurs	0 / 3	Nom variable incorrect.
Affichage final	0 / 1	-

NOTE FINALE : 06 / 20

Feedback :

- **Appréciation globale : Fragile.**
-

```
#include <stdio.h>
```

```
int main() {
```

```
int N, A, S, X, absents, presents;
```

```
printf("total students");
```

```
scanf("%d", &N);
```

```
printf("minimum attendance aquired");
```

```
scanf("%d", &A);
```

```
printf("absence threshold");
```

```
scanf("%d", &S);
```

```
while (i <= N) {
```

```
    scanf("%d", &X);
```

```
    printf("student = ", i);
```

```
    if (X < A) {
```

```
        scanf("%d", &absents);
```

```
    } else {
```

```
        scanf("%d", &presents);
```

```
    }
```

```
    } while (i <= N);
```

```
    printf("absents = %d", absents);
```

```
    printf("presents = %d", presents);
```

```
    if (absents >= S) {
```

```
        printf("session cancelled");
```

```
    } else {
```

```
        printf("session valid");
```

```
    }
```

```
    return 0;
```

```
}
```

Copy 7

```
#include <stdio.h>
int main()
{
    int N, A, S, X, absents, presents;
    printf("total students: ");
    scanf("%d", &N);
    printf("minimum attendance required: ");
    scanf("%d", &A);
    printf("absence threshold: ");
    scanf("%d", &S);
    while (i <= N)
    {
        scanf("%d", &x);
        printf("student = %d", i);
        if (x < A)
        {
            scanf("%d", &absents);
        }

        else
        {
            scanf("%d", &presents);
        }

    }

    printf("absents = %d", absents);
    printf("presents = %d", presents);
    if (absents >= S)
    {
        printf("session cancelled");
    }

    else
    {
        printf("session valid");
    }

    return 0;
}
```

Analyse :

Algorithmique :

- Boucle `while (i <= N)` mais `i` n'est **jamais initialisé** (valeur poubelle) ni incrémenté. Boucle possiblement infinie ou jamais exécutée.
- Lectures dans la boucle ? `scanf("%d", &absents)` -> l'utilisateur doit saisir le nombre d'absents ? Contresens de l'énoncé (on doit le calculer).

NOTE FINALE : 04 / 20

Feedback :

- **Appréciation globale : Très Insuffisant.** Incompréhension de l'énoncé.
-

```

#include <stdio.h>
int main() {
    int a, s, n, tot=0, pr=0, ab=0;
    printf("Enter N, A and S\n");
    scanf("%d %d %d", &n, &a, &s);
    int x=1;
    for(x=1; x<=n; x++) {
        if(x > a) {
            printf("present\n");
            pr = pr + 1;
        } else {
            printf("absent\n");
            ab = ab + 1;
        }
        if(ab >= s) { printf("cancelled\n"); break; }
        tot = tot + x;
    }

    printf("present students = %d\n", pr);
    printf("absent students = %d\n", ab);
    printf("total processed students = %d\n", tot);

    if(s < a) {
        printf("valid session\n");
    } else {
        printf("cancelled session\n");
    }
    return 0;
}

```

Copy 8

```
#include <stdio.h>
int main()
{
    int a, s, n, tot = 0, pr = 0, ab = 0;
    printf("Enter N, A and S\n");
    scanf("%d %d %d", &n, &a, &s);
    int x = 1;
    for (int x = 1; x <= n; x++)
    {
        if (x > a)
        {
            printf("present\n");
            pr = pr + 1;
        }

        else
        {
            printf("absent\n");
            ab = ab + 1;
        }

        if (ab >= s)
        {
            printf("cancelled\n");
            break;
        }

        tot = tot + 1;
    }

    printf("present students = %d\n", pr);
    printf("absent students = %d\n", ab);
    printf("total processed students = %d", tot);
    if (x > a)
    {
        printf("valid session\n");
    }

    else
    {
        printf("cancelled session\n");
    }

    return 0;
}
```


Analyse :

- Algorithmique :
- break utilisé (Interdit par l'énoncé mais fonctionnel).
 - Logique correcte.
 - Affichages complets.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	OK.
Initialisation	3 / 3	OK.
Condition boucle	3 / 4	Utilise break (interdit) au lieu condition composée.
Logique prés./abs.	4 / 4	OK.
Compteurs	3 / 3	OK.
Affichages inter.	2 / 2	OK.
Affichage final	1 / 1	OK.

NOTE FINALE : 19 / 20

Feedback :

- **Appréciation globale : Très Bon.** Code robuste. Pénalité minime pour le break.
-

Input reading: 3pts | Initialization: 3 pts | Loop condition: 4 pts | Counters logic: 4 pts | Stop conditions: 3 | Final output: 3

```

#include <stdio.h>
int main() {

    int N, A, S, X;
    int i, na=0, np;
    printf("Enter the total number of registered
    students");
    scanf("%d", &N);
    for (i=1, i <= N, i++) {
        printf("Student %d", i);
        scanf("%d", &X);
        if (X < A) {
            na = na + 1;
        }
        else {
            np = np + 1;
        }
    }
    if (i==N || np==S)
        printf("stop the simulation");
    printf("Present students is %d", np);
    printf("absent students is %d", na);
    if (na <= np)
        printf("valid");
    else
        printf("cancelled");
}

```

Copy 9

```
#include <stdio.h>
int main()
{
    int N, A, S, X;
    int i, na = 0, np;
    printf("Enter the total number of registered students: ");
    scanf("%d", &N);
    for (i = 1; i <= N; i++)
    {
        printf("student %d", i);
        scanf("%d", &X);
        if (X < A)
        {
            na = na + 1;
        }

        else
        {
            np = np + 1;
        }

        if (i == N || na == S)
        {
            printf("stop the simulation");
            break;
        }

    }

    printf("present students is %d", np);
    printf("absent students is %d", na);
    if (na <= np)
    {
        printf("valid");
    }

    else
    {
        printf("cancelled");
    }

    return 0;
}
```

Analyse :

Algorithmique :

- Initialisation `na=0` mais `np` non initialisé.
- `break` utilisé.
- Logic `if (i == N || na == S)`.
- Globalement ok sauf init `np`.

NOTE FINALE : 16 / 20

Feedback :

- **Appréciation globale : Bon.**
-

```

#include <stdio.h>
int main ( ) {
    int N, A, S, X, B = 0, P = 0;
    Printf ("enter total number of registered student");
    Scanf ("%d", &N);
    Printf ("enter minimum attendance required");
    Scanf ("%d", &A);
    Printf ("enter absence threshold");
    scanf ("%d", &S);
    for (i = 1; i <= N; i++) {
        Printf ("enter the number of attended session");
        Scanf ("%d", &X);
        if (X < A) {
            Printf ("student absent");
            B = B + 1;
        }
        Printf ("absent students = %d", B);
        else {
            Printf ("student Present");
            P = P + 1;
            Printf ("Present students = %d", P);
        }
        if (X < S) {
            Printf ("session cancelled");
        }
        else {
            Printf ("session valid");
        }
    }
    return 0;
}

```

Copy 10

```
#include <stdio.h>
int main()
{
    int N, A, S, X, B = 0, P = 0;
    printf("enter total number of registered student: ");
    scanf("%d", &N);
    printf("enter minimum attendance required: ");
    scanf("%d", &A);
    printf("enter absence threshold: ");
    scanf("%d", &S);
    for (i = 1; i <= N; i++)
    {
        printf("enter the number of attended session: ");
        scanf("%d", &X);
        if (X < A)
        {
            printf("student absent");
            B = B + 1;
            printf("absent students = %d", B);
        }

        else
        {
            printf("student present");
            P = P + 1;
            printf("present students = %d", P);
        }

        if (X < S)
        {
            printf("session cancelled");
        }

        else
        {
            printf("session valid");
        }

        return 0;
    }
}
```

Analyse :

Algorithmique :

- Logique correcte.
- Utilise compteurs B, P.
- Condition `if (X < S)` pour annulation ? Devrait être B (compteur absents) < S. Confusion variable.

NOTE FINALE : 13 / 20

Feedback :

- **Appréciation globale : Moyen.** Confusion sur la condition finale.
-

Input reading: 3pts | Initialization: 3 pts | Loop condition: 4 pts | Counters logic: 4 pts | Stop conditions: 3 | Final output: 3

```

#include <stdio.h>
int main() {
    int i;
    printf("entre the N");
    scanf("%d", &N);
    printf("entre the A");
    scanf("%d", &A);
    printf("entre the s");
    scanf("%d", &s);
    while(i <= N) { i++;
        printf("%d.", N); }
    if(i > A) then {
        printf("%d",
        printf("%d 'the present', i);
    }
    else
        printf("i > s of ");
    return 0;
}

```


Copy 11

```
#include <stdio.h>
int main()
{
    int i;
    printf("entre the N: ");
    scanf("%d", &N);
    printf("entre the A: ");
    scanf("%d", &A);
    printf("entre the S: ");
    scanf("%d", &S);
    while (i <= N)
    {
        i++;
        printf("%d", N);
        if (i > A)
        {
            printf("total present", i);
        }

        else
        {
            printf("S of", i);
        }

    }

    return 0;
}
```

Analyse :

Algorithmique :

- Boucle `while (i <= N)` avec `i` non initialisé.
- Pas de lecture de X (séances).
- Comparaisons incohérentes.

NOTE FINALE : 03 / 20

Feedback :

- **Appréciation globale : Très Insuffisant.**
-

Input reading: 3pts | Initialization: 3 pts | Loop condition: 4 pts | Counters logic: 4 pts | Stop conditions: 3 | Final output: 3

```

#include <stdio.h>
int main () {
    int N, A, s, x, student = 1, count = 0, absent, present, cancelled, valid;
    printf ("Enter the total number of registered students");
    scanf ("%d", &N);
    printf ("Enter minimum attendance required");
    scanf ("%d", &A);
    printf ("Enter absence threshold");
    scanf ("%d", &s);
    while (student <= N or count == s) {
        printf ("Enter number of attended sessions");
        scanf ("%d", &x);
        if (x < A) {
            printf ("Considered absent");
            count++;
        } else {
            printf ("Considered present");
            student++;
        }
        absent = count;
        present = N - count;
        printf ("number of present students : %d", present);
        printf ("number of absent student : %d", absent);
        if (present < absent) {
            printf ("Exam cancelled");
        }
        else {
            printf ("Exam valid");
        }
        return 0;
    }
}

```

Copy 12

```
#include <stdio.h>
int main()
{
    int N, A, S, X, student = 1, count = 0, absent, present;
    printf("Enter the total number of registered students: ");
    scanf("%d", &N);
    printf("Enter minimum attendance required: ");
    scanf("%d", &A);
    printf("Enter absence threshold: ");
    scanf("%d", &S);
    while (student <= N || count == 5)
    {
        printf("Enter number of attended sessions: ");
        scanf("%d", &x);
        if (x < A)
        {
            printf("Considered absent");
            count++;
        }

        else
        {
            printf("Considered Present");
            student++;
        }

    }

    absent = count;
    present = N - count;
    printf("number of present students: %d", present);
    printf("number of absent student: %d", absent);
    if (present < absent)
    {
        printf("Exam cancelled");
    }

    else
    {
        printf("Exam valid");
    }

    return 0;
}
```

Analyse :

Algorithmique :

- Condition `while` utilise `count == 5`. Pourquoi 5 ? Constante magique au lieu de `S`.
- Initialisation `student=1` OK.
- Logique correcte.

NOTE FINALE : 14 / 20

Feedback :

- **Appréciation globale : Moyen / Bon.**
-

```

#include <stdio.h>
int main() {
    int A, N, S, n = 0, m = 0;
    printf("entre the number A, N, S");
    scanf("%d %d %d", &A, &N, &S);
    for (i = 1; i <= N; i++) {
        scanf("%d", &n);
        if (n < A) {
            n++;
            printf("absent student: %d", n);
        }
        else {
            m++;
            printf("present student: %d", m);
        }
        if (m < S) {
            printf("امتحان صالح");
        }
        else {
            printf("امتحان ملغى");
        }
    }
    return 0;
}

```

Copy 13

```
#include <stdio.h>
int main()
{
    int A, N, S, n = 0, m = 0;
    printf("entre the number A, N, S: ");
    scanf("%d %d %d", &A, &N, &S);
    for (i = 1; i <= N; i++)
    {
        scanf("%d", &n);
        if (n < A)
        {
            n++;
            printf("absent student: %d", n);
        }

        else
        {
            m++;
            printf("present student: %d", m);
        }

        if (m < S)
        {
            printf("Exam valid");
        }

        else
        {
            printf("Exam cancelled");
        }

    }

    return 0;
}
```

Analyse :

Algorithmique :

- Boucle for sur N. Pas d'arrêt sur S.
- Logique correcte.
- Compteurs ok.

NOTE FINALE : 12 / 20

Feedback :

- **Appréciation globale : Moyen.**
-

Input reading: 3pts | Initialization: 3 pts | Loop condition: 4 pts | Counters logic: 4 pts | Stop conditions: 3 | Final output: 3

```

#include <stdio.h>
int main() {
    int N, A, x, m = 0, n = 0, S;
    printf("Enter the total number of registered student:");
    scanf("%d", &N);
    printf("Enter the minimum attendance required:");
    scanf("%d", &A);
    for (i = 1; i <= N; i++) {
        scanf("%d", &x);
        if (x < A) {
            m++;
            printf("absent student is : %d \n", m);
        } else {
            n++;
            printf("present student is : %d \n", n);
        }
        printf("Enter the absence threshold:");
        scanf("%d", &S);
        if (m < S) {
            printf("session valid");
        } else {
            printf("session cancelled");
        }
    }
    return 0;
}

```

Copy 14

```
#include <stdio.h>
int main()
{
    int N, A, sc, m = 0, n = 0, S;
    printf("entre the total number of registered student: ");
    scanf("%d", &N);
    printf("entre the minimum attendance required: ");
    scanf("%d", &A);
    for (i = 1; i <= N; i++)
    {
        scanf("%d", &x);
        if (x < A)
        {
            m++;
            printf("absent student is %d", m);
        }

        else
        {
            n++;
            printf("present student is %d", n);
        }

    }

    printf("entre the absence threshold: ");
    scanf("%d", &S);
    if (m < S)
    {
        printf("session valid");
    }

    else
    {
        printf("session cancelled");
    }

    return 0;
}
```

Analyse :

Algorithmique :

- Lectures correctes.
- Saisie S *après* la boucle !? (Impossible d'arrêter avant).
- Logique $m < S$ finale correcte.

NOTE FINALE : 12 / 20

Feedback :

- **Appréciation globale : Moyen.** Erreur ordre instructions.
-

Input reading: 3pts | Initialization: 3 pts | Loop condition: 4 pts | Counters logic: 4 pts | Stop conditions: 3 | Final output: 3

```

#include <stdio.h>
int main () {
    int N, A, S, i, x;
    int Sum1 = 0;
    int Sum2 = 0;
    printf("enter N: ");
    scanf("%d", &N);
    i = 1;
    while (i <= N || Sum1 != S) {
        printf("enter x: ");
        scanf("%d", &x);
        if (x < A) {
            printf("the student number %d: absent", i);
            Sum1 = Sum1 + 1;
        } else {
            printf("the student number %d: present", i);
            Sum2 = Sum2 + 1;
        }
        i++;
    }
    printf("%d, %d", Sum1, Sum2);

    if (Sum1 > Sum2) {
        printf("session cancelled");
    } else {
        printf("session valid");
    }

    return 0;
}

```

```
#include <stdio.h>
int main()
{
    int N, A, S, i, X;
    int Sum1 = 0;
    int Sum2 = 0;
    printf("enter N: ");
    scanf("%d", &N);
    i = 1;
    while (i <= N || Sum1 == S)
    {
        printf("enter X: ");
        scanf("%d", &X);
        if (X < A)
        {
            printf("the student num %d: absent", i);
            Sum1 = Sum1 + 1;
        }

        else
        {
            printf("the student num %d: present", i);
            Sum2 = Sum2 + 1;
        }

        i++;
    }

    printf("%d %d", Sum1, Sum2);
    if (Sum1 > Sum2)
    {
        printf("session cancelled");
    }

    else
    {
        printf("session valid");
    }

    return 0;
}
```

Analyse :

Algorithmique :

- Condition `while (i <= N || Sum1 == S)`. `||` au lieu de `&&` ? (Tant que pas fini OU seuil atteint... non ça devrait être `&&` non atteint). Avec `||`, si S atteint mais `i < N`, ça continue.
- Mais correct sur le reste.

NOTE FINALE : 15 / 20

Feedback :

- **Appréciation globale : Bon.**
-

Input reading: 3pts | Initialization: 3 pts | Loop condition: 4 pts | Counters logic: 4 pts | Stop conditions: 3 | Final output: 3

```
#include <stdio.h>
int main () {
    int N, A, S;
    printf("enter total number of registered students");
    scanf("%d %d %d", &N, &A, &S);
    for (A=1; A<=N; A++) {
        scanf("%d", &x);
        if (x < A) {
            printf("the student is absent");
        }
        else {
            printf("the student is present");
        }
        S++;
    }
    // ---- simulation stops ----
    if (
```

```
// ---- Final status ----
    if (S <= A) {
        printf("Session cancelled");
    }
    else {
        printf("session valid");
    }
}
```

```
return 0; }
```

Copy 16

```
#include <stdio.h>
int main()
{
    int N, A, S;
    printf("enter total number of registered students: ");
    scanf("%d %d %d", &N, &A, &S);
    for (A = 1; A <= N; A++)
    {
        scanf("%d", &x);
        if (x < A)
        {
            printf("the student is absent");
        }

        else
        {
            printf("the student is present");
        }

        count++;
    }

    // simulation stops if (S <= A)
    {
        printf("Session cancelled");
    }

    else
    {
        printf("session valid");
    }

    return 0;
}
```


Analyse :

Algorithmique :

- Déclaration for (A=1; A<=N; A++) : Utilise A (seuil min) comme compteur de boucle ! Écrase la valeur de A. Catastrophe.
- count non déclaré.

NOTE FINALE : 05 / 20

Feedback :

- **Appréciation globale : Très Insuffisant.** Erreur grave de réutilisation de variable.
-

Input reading: 3pts | Initialization: 3 pts | Loop condition: 4 pts | Counters logic: 4 pts | Stop conditions: 3 | Final output: 3

```

#include <stdio.h>
int main() {
    int N, A, S, X, i, n=0;
    int OS=0;
    int PS=0;
    printf("enter number de students \n");
    scanf("%d", &N);
    printf("enter minimum attendance required \n");
    scanf("%d", &A);
    printf("enter absence threshold \n");
    scanf("%d", &S);
    for (i=1; i <= N; i++) {
        if (i!=N && OS < S) {
            printf("enter number of attended sessions \n");
            scanf("%d", &X);
            if (X < A) {
                OS = OS + 1;
            }
            else {
                PS = PS + 1;
            }
            printf("%d\n", OS);
            printf("%d\n", PS);
            printf("%d\n", i);
            n = n + 1;
        }
    }

```

الحلقة

```

printf("%d\n", OS);
printf("%d\n", PS);
printf("%d\n", n);
return 0;
}

```

```

if (n < S) {
    printf("الامتحان ملغى \n");
}

```

```

else {
    printf("الامتحان صالح \n");
}

```

Copy 17

```
#include <stdio.h>
int main()
{
    int N, A, S, x, i, n = 0;
    int OS = 0;
    int PS = 0;
    printf("enter number de students\n");
    scanf("%d", &N);
    printf("enter minimum attendance required\n");
    scanf("%d", &A);
    printf("enter absence threshold\n");
    scanf("%d", &S);
    for (i = 1; i <= N; i++)
    {
        if (i <= N && OS < S)
        {
            printf("enter number of attended sessions\n");
            scanf("%d", &X);
            if (X < A)
            {
                OS = OS + 1;
            }

            else
            {
                PS = PS + 1;
            }

            printf("%d", OS);
            printf("%d", PS);
            printf("%d", i);
        }

        n = n + i;
    }

    if (n < S)
    {
        printf("Exam valid");
    }

    else
    {
        printf("Exam cancelled");
    }

    return 0;
}
```

Analyse :

Algorithmique :

- Condition `if (i <= N && OS < S)` à l'intérieur de la boucle `for` ? Redondant mais ok.
- Calcul `n = n + i` (somme des indices ?). Inutile.
- Variables `OS`, `PS` ok.

NOTE FINALE : 14 / 20

Feedback :

- **Appréciation globale : Moyen / Bon.**
-

```

#include <stdio.h>

int main() {
    int N, A, s, OS, PS;
    printf("Enter N and A and S: ");
    scanf("%d %d %d", &N, &S, &A);

    for (int i = 1; i <= N; i++) {
        int x;
        scanf("%d", &x);
        if (x < A) {
            printf("The student absent");
        }
        else {
            printf("The student present");
        }
        end if

        if (N < A (N < A)) {
            OS = PS + 1 & printf("Number student present");
        }
        else {
            PS = PS + 1 & printf("Number student absent");
        }
        if (N student are processed || absent student reaches) {
            printf("simulation stops");
        }
        end if

        if (N < A) {
            OS = OS + 1;
        }
        else {
            PS = PS + 1;
        }
        printf("%d %d %d", i, OS, PS);
        end if
    }

    if (OS = S)
        printf("session valid");
    else if
        printf("session Invalid");
    end if
    return 0;
}

```

```
#include <stdio.h>
int main()
{
    int N, A, S, OS, PS;
    printf("Enter N and A and S: ");
    scanf("%d %d %d", &N, &S, &A);
    for (int i = 1; i <= N; i++)
    {
        int n;
        scanf("%d", &n);
        if (n < A)
        {
            printf("The student absent");
        }

        else
        {
            printf("The student present");
        }

        if (n < A)
        {
            PS = PS + 1;
            printf("Number student present");
        }

        else
        {
            PS = PS + 1;
            printf("Number student absent");
        }

        if (N student are processed || absent student reaches)
        {
            printf("simulation stops");
        }

        if (0 < S = s)
        {
            printf("session valid");
        }

        else
        {
            printf("session Invalid");
        }
    }
}
```

```
    return 0;  
}
```

Analyse :

Algorithmique :

- Variable n déclarée dans la boucle (interdit en C89 strict mais ok C99).
- Logique correcte.
- Condition arrêt `if (N student processed || ...)` pseudo-code.
- Condition finale `if (OS = s)` (affectation au lieu de comparaison `==`).

NOTE FINALE : 10 / 20

Feedback :

- **Appréciation globale : Moyen -**. Erreurs de syntaxe C (`=`).